

9. Ignacio

Program Name: Ignacio.java

Input File: ignacio.dat

Ignacio was looking through a math puzzle book and encountered a very cool process to make a magic square. Magic squares, he learns, always have the same sum for any row, column, or main diagonal. The process described in his book works with squares of odd length and width, like 3 by 3, or 5 by 5, and so on. He needs your help to write a program to simulate this process so that he can identify any value in the square, given the row and column.

Here is an example of a magic square that is 3 by 3.

```
4 9 2
3 5 7
8 1 6
```

The top row sum is $4+9+2$, or 15. The sum for the leftmost column is $4+3+8$, also 15. Even the diagonals have a sum of 15: $4+5+6$ and $8+5+2$. "This is fascinating!", he thinks to himself.

The rules given in the puzzle book to build any size magic square, as long as it is an odd length and width, are as follows:

- Start with the value 1 in the middle column of the bottom row of the square.
- When possible, place the next number in the empty box down one row and across one column to the right.
- If the move described above in rule 2 is not possible because the box down one, right one is occupied, put the next number directly above the current box.
- If you have reached the bottom row, but not the last column, the next number goes to the top of the next column over.
- If you have reached the last column on the right, but not the bottom row, the next number goes to the first column of the next row down.
- If you have reached the bottom right corner of the square, put the next number directly above the current box.

Ignacio decides to build the 3 by 3 size box, shown above, following the rules.

1. He follows Rule 1 to place the 1 in row 3, column 2.
2. Since he's at the bottom row of the box, but not the last column, he follows Rule 4 and puts the 2 at the top of the next column.
3. Since he's now at the right column of the box, with no place to go "down one, across one", he follows Rule 5 and puts the 3 at the leftmost column of the next row down.
4. Since the box "down one, across one" is occupied by the 1, he puts the 4 directly above the 3, following Rule 3.
5. He puts the 5 and 6 using the "down one, across one" process described in Rule 2.
6. The 7 is placed using Rule 6, the 8 using Rule 5, and the 9 using Rule 4.

He now sees how the rules work, and can identify any value, given the row and column, such as row 3, column 2, which contains the 1, and row 2, column 1, which contains the 3.

Input - Several sets of three integer values, N, R, and C, all on one line, with single space separation.

Output - After building the magic square of size N, according to the rules listed and demonstrated above, output the value in row R, column C, of the completed magic square.

Sample data:

```
3 3 2
3 2 1
5 1 1
7 7 5
9 2 9
```

Sample Output:

```
1
3
11
10
25
```